

Syntaxe

Markdown is intended to be as easy-to-read and easy-to-write as is feasible. [Anonymous Contributors, "type|wiki|fr," (accessed April 1, 2019).](/fr/wiki/start)



Readability, however, is emphasized above all else. A Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. The single biggest source of inspiration for Markdown's syntax is the format of plain text email.

[3]: <http://daringfireball.net/projects/markdown/syntax> [4]: <http://michelf.ca/projects/php-markdown/extra/>

To this end, [Markdown's syntax][3] is comprised entirely of punctuation characters, which punctuation characters have been carefully chosen so as to look like what they mean. E.g., asterisks around a word actually look like `*emphasis*`. Markdown lists look like, well, lists. Even blockquotes look like quoted passages of text, as used in email. The original syntaxe is completed by the complimentary [Markdown Extra][4] which is in use here.

Emphasis

- Emphasize* `_emphasize_`
 - *Strong **Strong** *Emphasize* `_emphasize_` Strong **Strong** ## Other inline style
- Markdown is pretty limited but HTML is valid Markdown. Here is some useful HTML syntax. ### Striked text `<s>Striked text</s>` `<s>Striked text</s>` ### Underlined text `<u>Underlined text</u>` `<u>Underlined text</u>` ### Highlighted text `<mark>Highlighted text</mark>` `<mark>Highlighted text</mark>` ### Superscrit and subscript `H2O` `Au 20e siècle.` `H2O` `Au 20e siècle.` ### Keyboard With the `<kbd>cmd</kbd>+<kbd>s</kbd>` shortcut. With the `<kbd>cmd</kbd>+<kbd>s</kbd>` shortcut. ### Sample text Plain text style. Appear in monospace font. `<samp>Sample text</samp>` `<samp>Sample text</samp>` ## Liens ### External links A `[link](http://example.com "Title")`. A `[link](http://example.com "Title")`. ### Liens à la racine `[voici un lien](/syntax.md)` `[voici un lien](/syntax.md)` ### Liens dans la même section Par exemple, pour lier un terme du glossaire, si vous êtes dans le glossaire. `[voici un lien](syntax.md)` `[voici un lien](syntax.md)` ### Liens dans une autre section `[voici un lien](?id=en/glossary/eye.md)` `[voici un lien](?id=en/glossary/eye.md)` ou `[voici un lien](/dokuwiki/en/glossary/eye.md)` `[voici un lien](/dokuwiki/en/glossary/eye.md)` ### Referenced Links Some text with `[a link][1]` and another `[link][2]`. [1]: <http://example.com/> "Title" [2]: <http://example.org/> "Title" Some text with `[a link][5]` and another `[link][6]`. [5]: <http://example.com/> "Title" [6]: <http://example.org/> "Title" The reference section can be anywhere in the document ## Inline Images `![alt text](/dokuwiki/lib/exe/fetch.php?media=wiki:directwrite-comparison.png "title")` `![alt text](/dokuwiki/lib/exe/fetch.php?media=wiki:directwrite-comparison.png "title")` The "Alt" text (alternative text) makes images accessible to visually impaired You can use ``_media`` as a shorthand for ``lib/exe/fetch.php?media=``, so you only need to type ``![alt text](_media/namespace/image.png)`` `![alt`

`text](/dokuwiki/_media/wiki/directwrite-comparison.png) ![alt text](/dokuwiki/_media/wiki/directwrite-comparison.png) ##Referenced Images`
Smaller logo: `![Alt][1] [1]: /wp-smaller.png "Title" Smaller logo: ![Alt][7] [7]: /wp-smaller.png "Title" ##Linked Images` Wrap the entire image markup in a link: `![alt text](/dokuwiki/lib/exe/fetch.php?media=wiki:directwrite-comparison.png)](/dokuwiki/lib/exe/detail.php?media=wiki:directwrite-comparison.png) ![alt text](/dokuwiki/lib/exe/fetch.php?media=wiki:directwrite-comparison.png)](/dokuwiki/lib/exe/detail.php?media=wiki:directwrite-comparison.png)` the shorter form is `![alt text](/dokuwiki/_media/wiki/directwrite-comparison.png)](/dokuwiki/_media/wiki/directwrite-comparison.png) ![alt text](/dokuwiki/_media/wiki/directwrite-comparison.png)](/dokuwiki/_media/wiki/directwrite-comparison.png)` You can link to a page or the image itself. **##Footnotes** I have more `[^1]` to say up here. `[^1]`: To say down here. I have more `[^1]` to say up here. `[^1]`: To say down here. Footnotes will be added to the bottom of the document, with a link back to the original reference **##Line breaks** We do not support Markdown's typical double-space to generate a line break due to our built-in auto-linebreaking function. A regular line break will generate a line break on output. **##Bullet Lists** * Item * Item - Item + Item * Item * Item - Item + Item **##Numbered Lists** 1. Item 2. Item 1. Item 2. Item **##Mixed Lists** 1. Item 2. Item * Mixed * Mixed 3. Item 1. Item 2. Item * Mixed * Mixed 3. Item **##Blockquotes** > Quoted text. > > Quoted quote. > * Quoted > * List > Quoted text. > > Quoted quote. > * Quoted > * List **##Preformatted** Begin and end each line with two spaces or more to make text look e x a c t l y like you type i t. **##Code** ``This is code`` ``This is code`` **##Code block** `~~~~ This is a piece of code in a block ~~~~`
`~~~~ This is a piece of code in a block ~~~~` ``` This too ```` This too `````
##Syntax highlighting Pas pris en compte ```css #button { border: none; } ````css #button { border: none; } ````` It should work for ``actionscrip3`` — `▯`bash`` — ``clojure`` — ``coldfusion`` — ``cpp`` — ``csharp`` — ``css`` — ``delphi`` — ``erlang`` — ``fsharp`` — ``diff`` — ``groovy`` — ``html`` — ``javascript`` — ``java`` — ``javafx`` — ``matlab`` (keywords only) — ``objc`` — ``perl`` — ``php`` — ``text`` — ``powershell`` — ``python`` — ``r`` — ``ruby`` — ``scala`` — ``sql`` — ``vb`` — ``xml``. **##Headers** # Header 1 ## Header 2 {#header2} ### Header 3 {.head} #### Header 4 ##### Header 5 ##### Header 6 ##### # Header 1 ## Header 2 {#header2} ### Header 3 {.head} #### Header 4 ##### Header 5 ##### Header 6 ##### Closing hash marks are optional on all levels **##Definition Lists** Markdown : Text-to-HTML conversion tool Markdown : Text-to-HTML conversion tool Formatting for definition lists may vary between themes **##Abbreviations** Markdown converts text to HTML. ***[HTML]:** HyperText Markup Language Markdown converts text to HTML. ***[HTML]:** HyperText Markup Language Definitions can be anywhere in the document **##Tables** One | Two | Three — | :—: | —: Blue | White | Gray Green | Yellow | Red One | Two | Three | — | :—: | —: Blue | White | Gray | Green | Yellow | Red | The colon in the separator line tells the parser the alignment to apply to the column. Markdown inline is possible. **##Horizontal Rules** Three or more dashes or asterisks ``—`` ``*``

Citations

To make citations, use `<cite>` at the beginning of the citation and `</cite>` at the end of the citation.

Automatic Escaping for Special Characters

In HTML, there are two characters that demand special treatment: `<` and `&`. Left angle brackets are used to start tags; ampersands are used to denote HTML entities. If you want to use them as literal characters, you must escape them as entities, e.g. `<`, and `&`.

Ampersands in particular are bedeviling for web writers. If you want to write about 'AT&T', you need to write `AT&T`. You even need to escape ampersands within URLs. Thus, if you want to link to:

```
http://images.google.com/images?num=30&q=larry+bird
```

you need to encode the URL as:

```
http://images.google.com/images?num=30&amp;q=larry+bird
```

in your anchor tag `href` attribute. Needless to say, this is easy to forget, and is probably the single most common source of HTML validation errors in otherwise well-marked-up web sites.

Markdown allows you to use these characters naturally, taking care of all the necessary escaping for you. If you use an ampersand as part of an HTML entity, it remains unchanged; otherwise it will be translated into `&`.

So, if you want to include a copyright symbol in your article, you can write:

```
&copy;
```

©

and Markdown will leave it alone. But if you write:

```
AT&T
```

AT&T

Markdown will translate it to:

```
AT&amp;T
```

Similarly, because Markdown supports [inline HTML](#html), if you use angle brackets as delimiters for HTML tags, Markdown will treat them as such. But if you write:

```
4 < 5
```

Markdown will translate it to:

```
4 &lt; 5
```

However, inside Markdown code spans and blocks, angle brackets and ampersands are *always* encoded automatically. This makes it easy to use Markdown to write about HTML code. (As opposed to raw HTML, which is a terrible format for writing about HTML syntax, because every single `<` and `&` in your example code needs to be escaped.)

##Backslash Escapes

`http://example.com/` Markdown allows you to use backslash escapes to generate literal characters which would otherwise have special meaning in Markdown's formatting syntax. For example, if you wanted to surround a word with literal asterisks (instead of an HTML `` tag), you can use backslashes before the asterisks, like this:

```
\*literal asterisks\*
```

literal asterisks

Markdown provides backslash escapes for the following characters:

```
\  backslash  
`  backtick  
*  asterisk  
_  underscore  
{ } curly braces  
[ ] square brackets  
( ) parentheses  
#  hash mark  
+  plus sign  
-  minus sign (hyphen)  
.  dot  
!  exclamation mark
```

From:
<https://type.wiki/> - **type|wiki**

Permanent link:
<https://type.wiki/fr/syntax.md>

Last update: **16/03/2025 15:23**

